

Marco Tarini - Computer Graphics 2023/2024
Università degli Studi di Milano

La sequenza di trasformazioni nel rendering:
trasformazione di proiezione

Oggetto Mondo Vista Clip

1

Trasformazione di proiezione

M_M Model-Matrix

M_V View-Matrix

M_P Projection Matrix

Spazio Oggetto Spazio Mondo Spazio Vista Spazio Clip

2

Trasformazione di proiezione

✓ Problema classico:

⇒ in arte, architettura, ...

Alcune delle soluzioni classiche hanno un nome come...

✓ come riportare

⇒ oggetti 3D

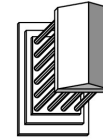
⇒ su un
piano 2D



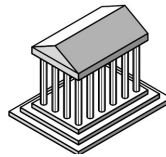
Prospetto Frontale



Assonometria cavaliera



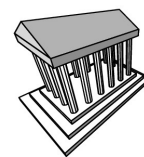
Pianta Obliqua



Assonometria
Isometrica



Prospettiva a
punto di fuga singolo



Prospettiva a punto
di fuga triplo

3

Trasformazione di proiezione “ortogonale” (oppure “ortografica”)

✓ Un modo banale per proiettare da 3D a 2D? (“proiettare” = perdere una dimensione)

⇒ facile: azzerare la z

⇒ matrice corrispondente:

$$P_Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4

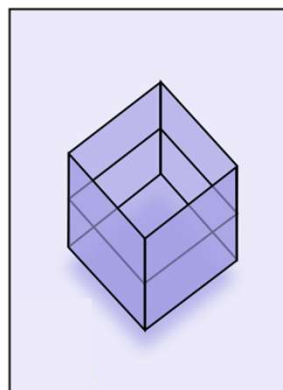
Trasformazione di proiezione "ortogonale" (o "ortografica")

- ✓ Abbiamo ottenuto una *proiezione ortogonale*
 - ⇒ non c'è alcuna prospettiva:
la dimensione (a schermo) degli oggetti *non* dipende dalla loro distanza dall'osservatore
 - ⇒ linee parallele (in spazio vista)
rimangono parallele anche a schermo
 - ⇒ la direzione di vista è costante in ogni pixel
 - ⇒ Approssima bene una situazioni reali come:
 - un cannocchiale molto potente
che inquadri la scena da molto lontano
 - una vista satellitare di un paesaggio urbano
 - Insomma, lunghezza focale molto lunga

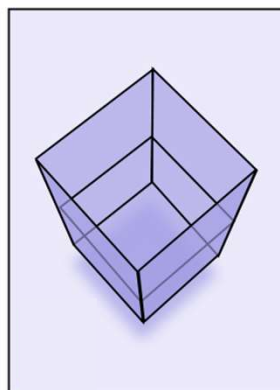


6

Differenze



Proiezione
ortografica



Proiezione
prospettica

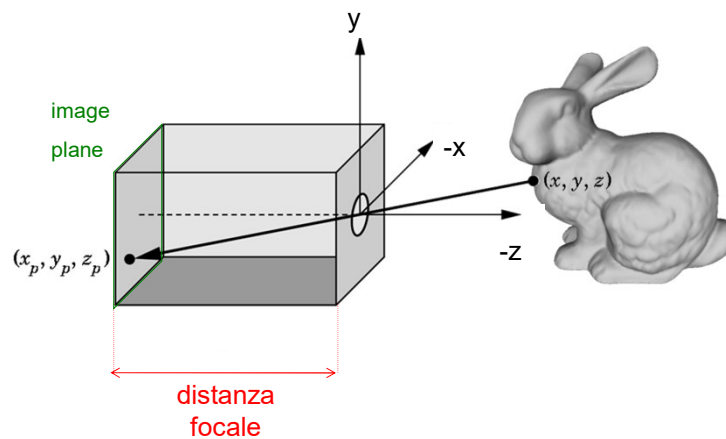
Vediamo ora come ottenere una **matrice di proiezione prospettica**



8

Il nostro modello semplificato di camera (richiamo)

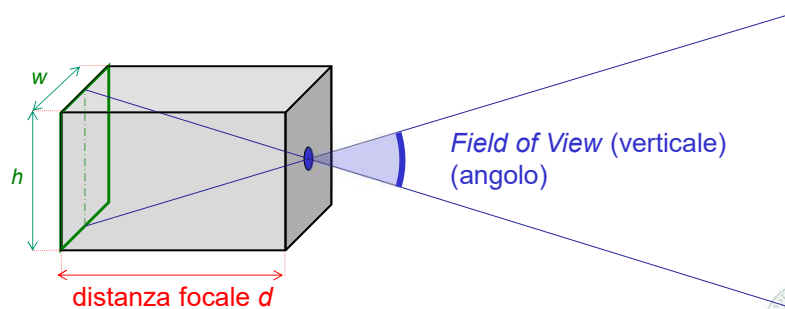
✓ pin-hole camera



11

Parametri *intrinseci* della camera: (solo i principali)

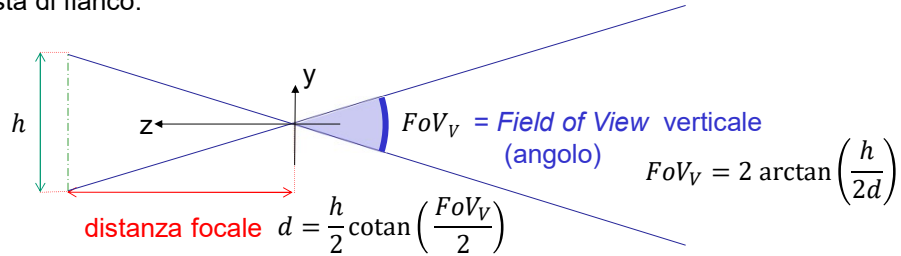
- ✓ dimensioni image plane (w, h) (in spazio vista)
- ✓ distanza focale (d) (in spazio vista) *oppure* Field of View (FoV)
 - ⇒ si possono ottenere uno dall'altro, sapendo h (come?)
 - ⇒ il valore d è facile da usare nei conti, ma FoV è intuitivo da determinare:
 - FoV grande $>60^\circ$ (dist foc. piccola): «grandangolo»
 - FoV piccolo $<45^\circ$ (dist foc. grande): «teleobiettivo»



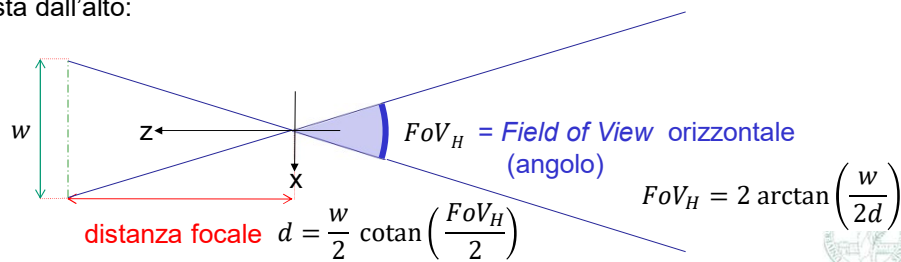
12

Da Field of View (verticale o orizzontale) a Distanza focale, o viceversa

Vista di fianco:



Vista dall'alto:

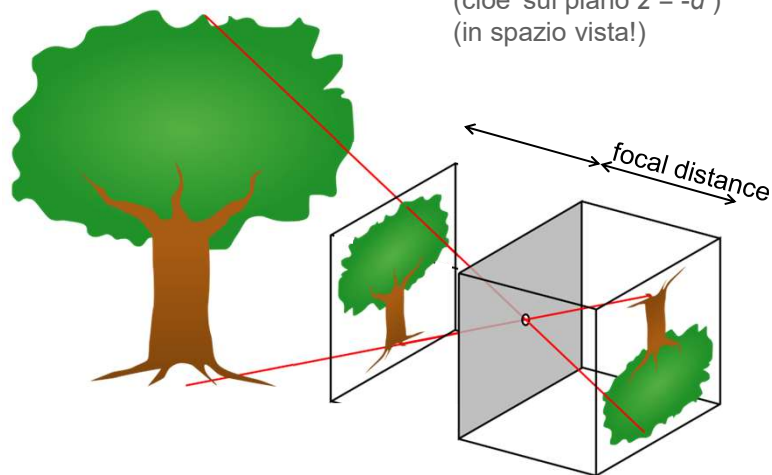


14

Pin Hole camera (richiamo)

per semplicità, immaginiamoci il piano immagine *davanti* alla camera (e nn ribaltato), piuttosto che *dietro* (e ribaltato).

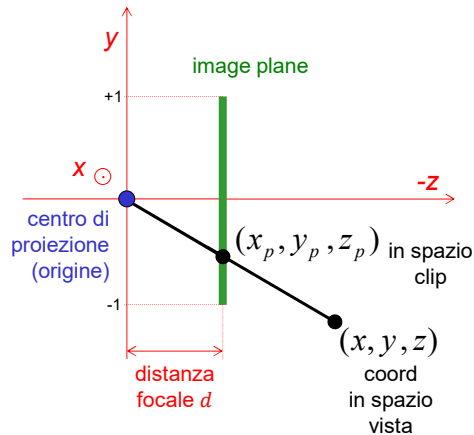
(cioè sul piano $z = -d$)
(in spazio vista!)



15

Matematicamente...

(ipotizzando per ora che le dimensioni della camera siano $w=2$ e $h=2$)



$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = k \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{con } k \text{ t.c. } z_p = -d$$

quindi... $k = -d / z$

$$\text{e } \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = \begin{pmatrix} -d \cdot x / z \\ -d \cdot y / z \\ -d \end{pmatrix}$$



16

Trasformazione di proiezione

✓ Dunque la nostra **trasformazione di proiezione**, sui **punti**, è questa:

$$f_P \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -d \cdot x / z \\ -d \cdot y / z \\ -d \end{pmatrix}$$

$\uparrow$ coord cartesiane in spazio vista $\uparrow$ coord cartesiane in spazio clip

✓ Posso esprimerla in forma matriciale? Cioè come...

$$\mathbf{M}_P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -d \cdot x / z \\ -d \cdot y / z \\ -d \\ 1 \end{bmatrix}$$

$\nwarrow$ Matrice di proiezione $\uparrow$ coord omogenee in spazio vista $\uparrow$ coord omogenee in spazio clip

(nota: non mi serve applicare questa trasformazione a **vettori**)



17

Estendiamo la rappresentazione di punti e vettori in coordinate **omogenee**

Punti:

$$\neq 0 \rightarrow \mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ * \end{bmatrix}$$

Vettori:

$$0 \rightarrow \vec{v} = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$



19

Estendiamo la notazione

✓ Posso esprimere i *punti* anche con la notazione

$$\mathbf{p} = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \quad \text{con } w \neq 0$$

$$\begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \xrightarrow{\text{divisione per 4ta comp}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

anche detta
normalizzazione affine



20

Estendiamo la notazione

✓ Per es:

$$\begin{bmatrix} 20 \\ 10 \\ 30 \\ 10 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 3 \\ 1 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.1 \\ 0.3 \\ 0.1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 6 \\ 2 \end{bmatrix} \dots$$

questa qui
è in "forma
canonica"
($w = 1$)

sono alcune delle
coordinate omogenee
del **punto**
che ha queste
coordinate cartesiane

$$\begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

punto

$$\begin{bmatrix} 2 \\ 1 \\ 3 \\ 0 \end{bmatrix}$$

sono le uniche
coordinate omogenee
del **vettore**
che ha queste
coordinate cartesiane

$$\begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$$

vettore



21

Coordinate omogenee di punti (riassunto)

- ✓ Il punto P di coordinate **cartesiane** (x, y, z) è rappresentato in coordinate **omogenee** come (xw, yw, zw, w) , con w qualsiasi, eccetto 0
- ✓ Coordinate omogenee diverse (x, y, z, w) e (x', y', z', w') possono rappresentare lo stesso punto;
⇒ quando? Sapresti scrivere le condizioni in cui questo succede?
- ✓ Quando $w = 1$ (forma **canonica**) le coord. cartesiane del punto coincidono con le prime tre coord. omogenee.
- ✓ Con $(x, y, z, w \neq 0)$ si rappresentano **punti**, con $(x, y, z, 0)$ si rappresentano **vettori**.



22

Coordinate omogenee (o affini)

✓ Tutte le **matrici di trasformazione** che abbiamo visto fin'ora continuano a "funzionare" anche su **punti** espressi in coordinate omogenee non canoniche (cioè con $w \neq 1$)

✓ Per esempio, traslazione:

$$\begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 20 \\ 15 \\ 50 \\ 10 \end{pmatrix} = \begin{pmatrix} 70 \\ 35 \\ 80 \\ 10 \end{pmatrix}$$

Matrice di traslazione di (+5, +2, +3)
Punto di coord cartesiane (2, 1.5, 5)
Punto di coord cartesiane (7, 3.5, 8)

(+5, +2, +3)

✓ Ora possiamo esprimere anche la **proiezione prospettica**...

23

Matrice di proiezione prospettica

matrice della
proiezione prospettica
(nota: non è affine,
perché l'ultima riga
non è 0,0,0,1)

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix}$$

$$\mathbf{P} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ -z/d \end{pmatrix} \rightarrow \begin{pmatrix} -x \cdot d/z \\ -y \cdot d/z \\ -d \end{pmatrix}$$

...che esprime il punto di coordinate cartesiane... (divido per w)

24

Matrice di proiezione prospettica: alternativa (scegliamo arbitrariamente questa)

matrice di trasformazione
per la
proiezione prospettica
(nota: non è affine)

$$\mathbf{P} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\mathbf{P} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ d \cdot z \\ -z \end{pmatrix}$$

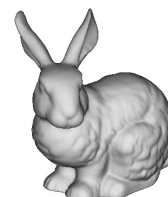
...che esprime il
punto di coordinate
cartesiane...
(divido per w)

$$\begin{pmatrix} -x \cdot d/z \\ -y \cdot d/z \\ -d \end{pmatrix}$$



25

Proiezione Prospettica: che effetto fa



d piccolo

$$\mathbf{P} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

d grande

Più distorsione
prospettica.

Effetto "fish-eye"
(grandangolo)

Proporzioni
più mantenute

Effetto "zoom"
(eg. vista dal
satellite)



27

Non è una buona idea perdere l'informazione sulla z (cioè la distanza del punto dal POV)

La lunghezza focale, cioè la profondità della pin-hole camera (ipotizzando che la sua altezza e larghezza siano 2)

normalizzazione affine (per l'estrazione di coordinate Cartesiane)

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ d \cdot z \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} -x \cdot d/z \\ -y \cdot d/z \\ -d \\ 1 \end{pmatrix}$$

perspective matrix view coords clip coords

PROBLEMA: il valore di z risulta **costante**.
Formalmente è corretto
(dato che il punto finisce sempre sul piano immagine)
ma perdiamo una dimensione inutilmente



28

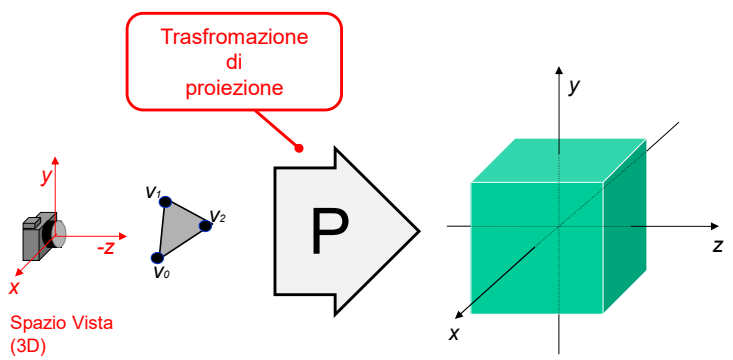
In realtà la trasf. di Proiezione non deve perdere la terza dimensione (la z): ci sarà utile

Trasformazione di proiezione

Spazio Vista (3D)

Spazio Clip [ancora 3D!]

La parte visibile casca per definizione in $[-1,+1] \times [-1,+1] \times [-1,+1]$
(indipendentemente dalla dimensione, aspect ratio, o risoluzione dello schermo).
(per questo le coord in spazio clip sono dette anche "Normalized Device Coordinates", NDC)




30

Modificare la matrice di proiezione per produrre una z in spazio clip fra -1 e +1

In Spazio Vista:

In spazio Clip:

31

Dettaglio tecnico

32

Dettaglio tecnico

Nell'algoritmo di rendering che stiamo descrivendo, la fase di **processing per vertice** ha il compito di produrre ...

✓ (a partire dalle coordinate in **spazio oggetto**)

le **coordinate clip** del vertice

sotto forma di **coordinate omogenee**

$(x \cdot w, y \cdot w, z \cdot w, w)$

✓ Se le coord x, y, z sono in $[-1..+1]$, il vertice è inquadrato

✓ quando x, y , **oppure** z fuori da $[-1..+1] \rightarrow$ fuori da immagine!

✓ cioè: se $z > +1$, troppo lontano: non disegno

cioè: se $z < -1$, troppo vicino: non disegno

✓ La matrice di proiezione deve quindi occuparsi anche di produrre anche la z corretta in spazio clip



33

Modificare la matrice di proiezione per rimappare la Z correttamente

✓ Scelgo arbitrariamente...

$\Rightarrow z_near$: la distanza dalla camera della cosa più *vicina* da inquadrare nell'immagine

$\Rightarrow z_far$: la distanza dalla camera della cosa più *lontana* da inquadrare nell'immagine

$\Rightarrow$ nota: entrambe sono distanze definite in spazio vista

- sono due reali > 0

✓ La trasf di proiezione deve portare l'intervallo Z (in spazio vista) $[-z_near .. -z_far]$ nell'intervallo Z (in spazio clip) $[-1..+1]$



34

Modificare la matrice di proiezione per rimappare la Z correttamente

- ✓ Voglio modificare P in modo che la z in output non sia costante (ma dipenda dalla z in input)

$$\begin{array}{c} \text{MATRICE PROIEZIONE} \end{array} \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{array}{c} \text{COORDINATE VISTA OMOGENEE} \\ \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \end{array} = \begin{array}{c} \text{COORDINATE CLIP OMOGENEE} \\ \begin{pmatrix} d \cdot x \\ d \cdot y \\ Az + B \\ -z \end{pmatrix} \end{array} \rightarrow \begin{array}{c} \text{COORD. CLIP OMOGENEE, MA IN FORMA CANONICA} \\ \begin{pmatrix} -x \cdot d/z \\ -y \cdot d/z \\ -A - B/z \\ 1 \end{pmatrix} \end{array}$$

quali valori devo usare per A e B ?

view coords

clip coords

35

Modificare la matrice di proiezione per produrre una z in spazio clip fra -1 e +1

- ✓ Troviamo i valori per A e B
- ✓ Vogliamo che:
 - ⇒ se $z_{\text{vista}} = -z_{\text{Near}}$, allora z_{clip} deve essere -1
 - ⇒ se $z_{\text{vista}} = -z_{\text{Far}}$, allora z_{clip} deve essere $+1$
 - ⇒ Quindi

$$-A + \frac{B}{z_{\text{Near}}} = -1 \quad -A + \frac{B}{z_{\text{Far}}} = +1$$

- ✓ Risolvo per A e B ottenendo... (verificare!)

$$A = \frac{z_{\text{Far}} + z_{\text{Near}}}{z_{\text{Far}} - z_{\text{Near}}} \quad B = \frac{2 \cdot z_{\text{Far}} \cdot z_{\text{Near}}}{z_{\text{Far}} - z_{\text{Near}}}$$

36

Matrice di Proiezione Prospettica ottenuta

$$P = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & \frac{(z_F + z_N)}{(z_F - z_N)} & \frac{2 \cdot z_F \cdot z_N}{(z_F - z_N)} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

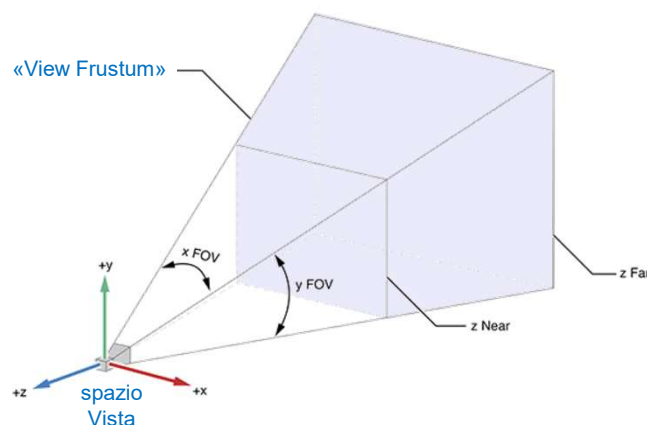
d lunghezza focal

(se H e W della pin-hole camera sono 2 e 2)



37

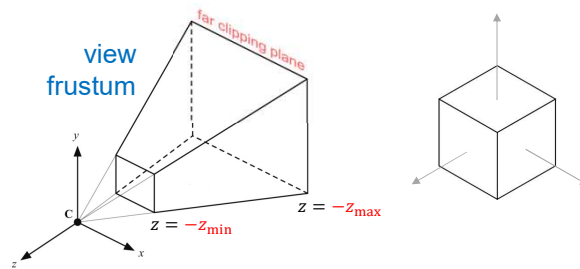
View Frustum: la zona dello spazio vista che contiene ciò che viene visto dalla camera



39

Limiti dello spazio visibile (cioè quello inquadrato dalla camera)

	in Spazio Vista	in Spazio Clip
x	«left clipping plane»	> -1
	«right clipping plane»	$< +1$
y	«bottom clipping plane»	> -1
	«top clipping plane»	$< +1$
z	$< -z_{\min}$	> -1
	$> -z_{\max}$	$< +1$



40

Spazio screen (screen space) (anche detto spazio viewport)

✓ **Viewport:** area dello schermo in cui renderizziamo l'immagine

⇒ per es: nelle app “a schermo intero”,
il viewport è tutto lo schermo

✓ **Spazio “screen”**

⇒ Uno spazio in cui i pixel sono a coordinate intere

⇒ Asse x : verso la destra dello schermo
va da 0 a risoluzione orizzontale del *viewport*

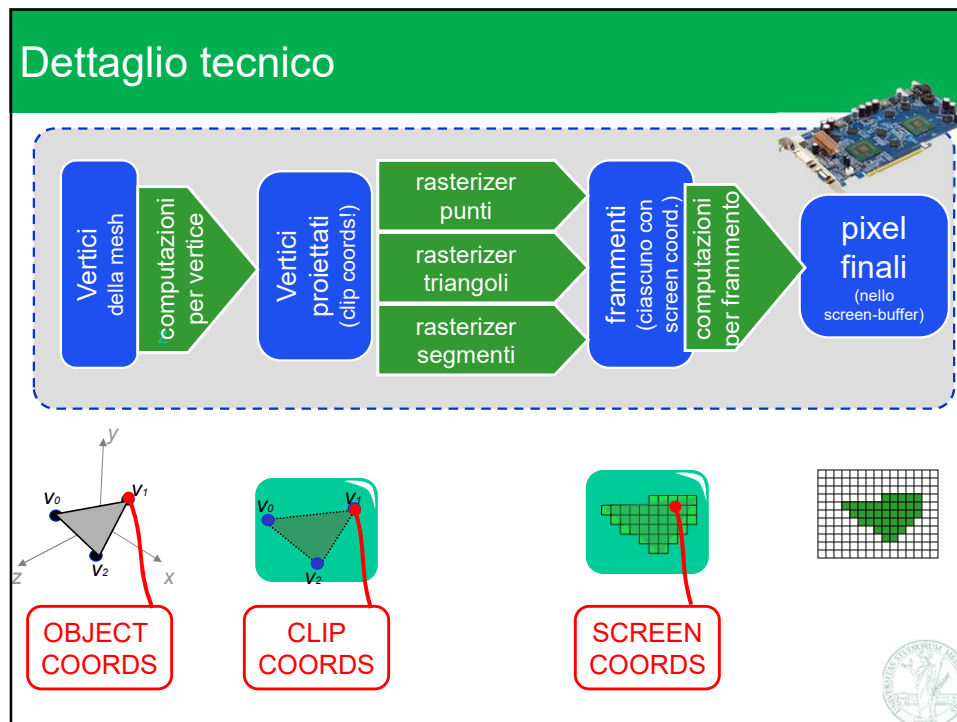
⇒ Asse y : verso l'alto dello schermo
va da 0 a risoluzione verticale del *viewport*

⇒ Punto 0,0: punto in basso a sx dello schermo

⇒ Asse z : direzione profondità del pixel
la coordinata z è un valore da 0 a 1 detto *depth*



42



43

Dettaglio tecnico

La fase di **rasterizzatore** del pipeline

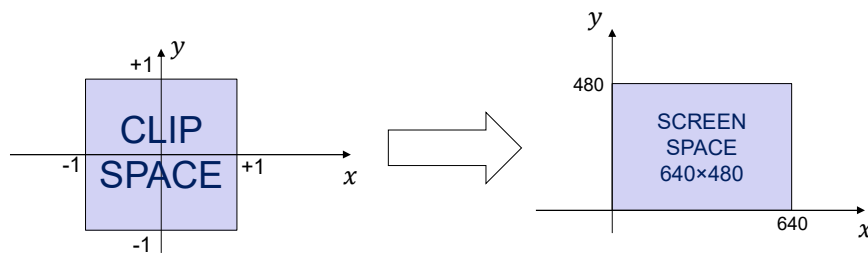
- ✓ riceve i vertici dalla fase precedente in **coordinate clip omogenee** (4 numeri *reali*)
- ✓ come preliminare: converte quindi queste coordinate...
 - ⇒ da: omogenee in *clip space*
 - ⇒ a: cartesiane in *clip space* (dividendo per w)
 - ⇒ a: cartesiane in *screen space* (mapping lineare, vedi dopo)
- ✓ produce poi frammenti a **coordinate screen cartesiane** (e intere)
 - ⇒ e.s. in $[0..639] \times [0..479]$, se lo schermo è 640 x 480
 - ⇒ vedi lezione sulla rasterizzazione

44

Da Clip Space a Screen Space (o "Viewport")

✓ In **coordinate screen**,
i pixel (e i frammenti)
sono quadratini
a coordinate intere
di dimensione 1×1

il rettangolo di schermo
che contiene il rendering
(nelle applicazioni
che non sono a schermo intero)



45

Trasformazione di viewport (automatica): da Clip Space a Screen Space (o Viewport)

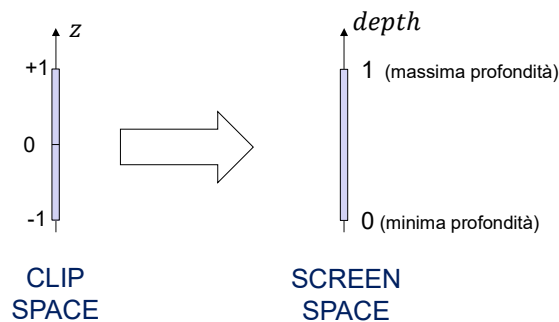
$$f_{VIEWPORT} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (x+1)/2 \cdot RES_X \\ (y+1)/2 \cdot RES_Y \end{pmatrix}$$

in
[-1,+1] × [-1,+1]
in
[0,1] × [0,1]

47

La Z in spazio Screen (convenzione)

- ✓ La coordinata cartesiana z in spazio clip (che va da -1 a +1 per le cose visibili) viene anch'essa trasformata in spazio screen (o coordinate pixel), dove prende in nome di (pixel) *depth* (profondità del pixel), e va nell'intervallo 0,1



48

Da Clip Space a Screen Space (trasformazione di viewport)

$$f_{to_screen} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (x+1)/2 \cdot RES_X \\ (y+1)/2 \cdot RES_Y \\ (z+1)/2 \end{pmatrix}$$

in
in
 $[-1, +1]^3$
 $[0, 1]^3$

Quando $RES_X \neq RES_Y$ questo passaggio corrisponde ad una scalatura non uniforme della x e y

49

Limiti dello spazio visibile (sommario)

	Spazio Vista	Spazio Clip	Spazio Schermo	
x	«left clipping plane»	> -1	≥ 0	fragment coordinates
	«right clipping plane»	$< +1$	$< res_x$	
y	«bottom clipping plane»	> -1	≥ 0	
	«top clipping plane»	$< +1$	$< res_y$	
z	$< -z_{min}$	> -1	≥ 0	fragment depth
	$> -z_{max}$	$< +1$	≤ 1	

50

Trasformazione di viewport : include uno scaling NON uniforme (distorce)

- ✓ Il quadrato in Spazio Clip: $[-1, +1] \times [-1, +1] \dots$
 $\Rightarrow$ un quadrato (di lato 2)
- ✓ $\dots$ diventa il Viewport: $[0, resX] \times [0, resY]$
 $\Rightarrow$ è quadrato solo se $resX = resY$
- ✓ Il passaggio da Clip Space a Screen Space introduce uno scaling NON uniforme (se $resX \neq resY$)

quindi:

- ✓ nella Trasformazione di Proiezione deve compensare questo effetto applicando uno scaling NON uniforme inverso

51

Pre-correzione dell'aspect ratio

✓ Matrice di proiezione prospettica con correzione dell'aspect ratio $a = \frac{RES_X}{RES_Y}$

$$P = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & da & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix} \text{ oppure } P = \begin{pmatrix} d/a & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Nota: la trasformaz. di proiezione dipende dai parametri INTRINSECI della macchina fotografica... compreso l'aspect ratio del suo frame!



52

Matrice di Proiezione Prospettica finale

$$P = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & a \cdot d & 0 & 0 \\ 0 & 0 & \frac{(Z_F + Z_N)}{(Z_F - Z_N)} & \frac{2 \cdot Z_F \cdot Z_N}{(Z_F - Z_N)} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

tutto in spazio vista

- a aspect ratio
- d lunghezza focale
- Z_N z-near
- Z_F z-far

Sono i parametri intrinseci!



53